Modified Block BiCGSTAB for Lattice QCD

Y. Nakamura[a], K. -I. Ishikawa[b], Y. Kuramashi[c,d,a,f], T. Sakurai[e,f], H. Tadano[e,d,f]

[a]*RIKEN Advanced Institute for Computational Science, Kobe, Hyogo 650-0047, Japan*
[b]*Graduate School of Science, Hiroshima University, Higashi-Hiroshima, Hiroshima 739-8526, Japan*
[c]*Graduate School of Pure and Applied Sciences, University of Tsukuba, Tsukuba, Ibaraki 305-8571, Japan*
[d]*Center for Computational Sciences, University of Tsukuba, Tsukuba, Ibaraki 305-8577, Japan*
[e]*Department of Computer Science, University of Tsukuba, Tsukuba, Ibaraki 305-8573, Japan*
[f]*JST, CREST, 5, Sanbancho, Chiyoda-ku, Tokyo 102-0075, Japan*

Abstract

We present results for application of block BiCGSTAB algorithm modified by the QR decomposition and the SAP preconditioner to the Wilson-Dirac equation with multiple right-hand sides in lattice QCD on a $32^3 \times 64$ lattice at almost physical quark masses. The QR decomposition improves convergence behaviors in the block BiCGSTAB algorithm suppressing deviation between true residual and recursive one. The SAP preconditioner applied to the domain-decomposed lattice helps us minimize communication overhead. We find remarkable cost reduction thanks to cache tuning and reduction of number of iterations.

Keywords: Lattice gauge theory, Lattice Dirac equation, Multiple right-hand sides, Block Krylov subspace, Domain decomposition

1. Introduction

Lattice QCD simulations initiated 30 years ago stand finally at the point where one can obtain results of physical observables at the physical up, down and strange quark masses [1]. The next steps would be refinement of the results reducing the systematic errors and challenge to computationally difficult problems, e.g. calculation of disconnected diagrams. A main difficulty in lattice QCD simulations is that solution of Dirac equation, which have to be repeated many times both in configuration generation and measurement of physical observables on given configurations, is computationally expensive near the physical up and down quark masses. In the measurement of physical observables, however, computational cost may be reduced by block Krylov subspace methods [2], since its expensive part is the multiple right-hand side problem. (This is not the case for configuration generation.) One can expect that block Krylov subspace methods make convergence faster with the aid of better search vectors generated from wider Krylov subspace enlarged by number of right-hand side vectors in comparison with non-blocked method. Another possible ingredient to improve performance is an efficient use of memory bandwidth in implementation of block matrix-vector multiplication.

Since the Dirac matrix in lattice QCD is non-Hermitian, we might expect the block BiCGSTAB algorithm [3] is applicable in a straightforward way. One problem in block Krylov subspace methods, however, is that the true residual stops decreasing at some point, while the recursive one continues to decrease. Recently, three of the authors have proposed a new algorithm named block BiCGGR, which showed significant improvement for this problem [4, 5, 6].

In this paper we improve block BiCGSTAB algorithm with two modifications. First one is the QR decomposition, which is known to improve the numerical accuracy in block CG [2, 7] and also useful for block BiCGSTAB algorithm [8]. Second one is Schwarz alternating procedure (SAP) preconditioner proposed by Lüscher [9], which is applied to the domain-decomposed lattice. We can minimize communication overhead with the SAP preconditioner.

This paper is organized as follows. In Sec. 2 we explain the algorithmic details of the modified block BiCGSTAB with SAP preconditioner. We present the results of the numerical test in Sec. 3. Conclusions and discussions are summarized in Sec. 4.

2. Algorithm

2.1. *Modified Block BiCGSTAB*

We consider to solve the linear systems with the multiple right-hand sides expressed as

$$AX = B\,, \tag{1}$$

where A is an $N \times N$ complex sparse non-Hermitian matrix. X and B are $N \times L$ complex rectangular matrices given by

$$X = \left(\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(i)}, \ldots, \boldsymbol{x}^{(L)}\right), \tag{2}$$
$$B = \left(\boldsymbol{b}^{(1)}, \ldots, \boldsymbol{b}^{(i)}, \ldots, \boldsymbol{b}^{(L)}\right). \tag{3}$$

In the case of the Wilson-Dirac equation the matrix dimension is given by $N = L_x \times L_y \times L_z \times L_t \times 3 \times 4$ with $L_x \times L_y \times L_z \times L_t$ the volume of a hypercubic four-dimensional lattice. L is the number of the right-hand-side vectors which is called source vectors in lattice QCD. L is 12 in the simplest case and $O(10) - O(100)$ (perhaps more in some case) for the stochastic method.

The matrix-vector multiplication for the Wilson-Dirac equation is written as

$$A\phi = \sum_{x=1}^{L_x \times L_y \times L_z \times L_t} (\phi_x - \kappa\eta_x)\,, \tag{4}$$

$$\eta_x = \sum_{\mu=1}^{4} \left[(1-\gamma_\mu)U_{x,\hat{\mu}}\phi_{x+\hat{\mu}} + (1+\gamma_\mu)U^\dagger_{x-\mu,\hat{\mu}}\phi_{x-\hat{\mu}}\right], \tag{5}$$

where ϕ_x and η_x contain 12 complex numbers at site x, γ_μ are the gamma matrices, $U_{x,\hat{\mu}}$ are link variables of SU(3) matrix and κ is hopping parameter. Computation of η_x requires 1320[1] floating point number operations and 360 words per site. This means the value of Flops/Byte is about 0.9 (0.45) in the single (double) precision. It should be difficult to obtain high performance on recent computer architecture.

In block Krylov subspace methods, Eq. (5) can be expressed as

$$\boldsymbol{\eta}_x^{(i)} = \sum_{\mu=1}^{4} \left[(1-\gamma_\mu)U_{x,\hat{\mu}}\boldsymbol{\phi}^{(i)}_{x+\hat{\mu}} + (1+\gamma_\mu)U^\dagger_{x-\mu,\hat{\mu}}\boldsymbol{\phi}^{(i)}_{x-\hat{\mu}}\right], \tag{6}$$

with $i = 1, \ldots, L$. An important point is that same 8 link variables around site x are used in common for $\phi^{(i)}$ with $i = 1, \ldots, L$ and their size is just 576 (1152) bytes in the single (double) precision, which are small enough to be retained in low level cache, for example L1 cache (if there is). This allows us more efficient usage of cached data than repeating L independent matrix-vector multiplications. Figure 1 illustrates how Flops/Byte increases as L is enlarged. For an effective use of cache, we arrange loop for the index of i ($i = 1, \ldots, L$) in the most inner level with i running first in memory allocation for vectors.

[1] 1296 flops if γ_μ is non-relativistic representation.

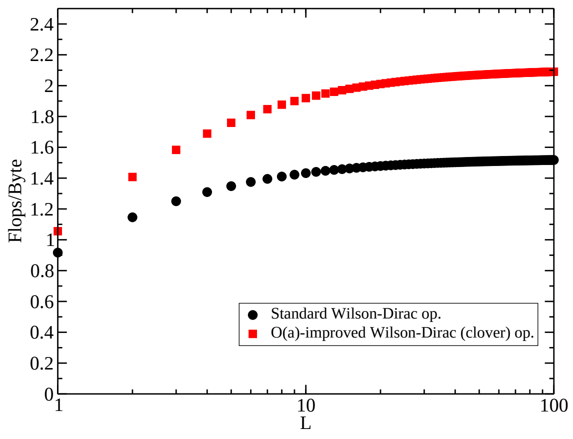

Figure 1: Flops/Byte as a function of L for standard (black circle) and $O(a)$-improved (red square) Wilson-Dirac operators in the single precision.

Pseudocode for modified block BiCGSTAB algorithm is described in Algorithm 2.1. Note that preconditioner M at lines 4.2 and 4.6 in Algorithm 2.1 must be implemented by a stationary iterative method in this algorithm since the common preconditioning should be applied to all right-hand sides, though nonstational iterative methods are often used in lattice QCD [10]. The orthogonalization of P improves numerical accuracy since each span works effectively to search approximated solutions. We employ modified Gram-Schmidt method for the QR decomposition. Even when non-block BiCGSTAB fails to converge, modified block BiCGSTAB may converge by adding dummy right-hand side vectors if they can play a supplementary role [8]. We also present a memory saving version in Algorithm 2.2.

2.2. *Preconditioning*

In this work we employ the $O(a)$-improved [2] Wilson fermions. After Jacobi preconditioning (division of $I +$

[2] The leading cut-off error in terms of the lattice spacing a is removed.

Algorithm 2.1: MODIFIED BLOCK BICGSTAB(A, M, B, ϵ)

```
1    initial guess $X \in \mathbb{C}^{N\times L}$
2    $R = B - AX$
3    $P = R$
4    choose $\tilde{R} \in \mathbb{C}^{N\times L}$
while  $\max_i(|r^{(i)}|/|b^{(i)}|) \le \epsilon$
do
     4.1   QR decomposition $P = Q\gamma,\ P \leftarrow Q$
     4.2   $U = MP$
     4.3   $V = AU$
     4.4   solve $(\tilde{R}^H V)\alpha = \tilde{R}^H R$ for $\alpha$
     4.5   $T = R - V\alpha$
     4.6   $S = MT$
     4.7   $Z = AS$
     4.8   $\zeta = \mathrm{Tr}(Z_k^H T_k)/\mathrm{Tr}(Z_k^H Z_k)$
     4.9   $X \leftarrow X + U\alpha + \zeta S$
     4.10  $R = T - \zeta Z$
     4.11  solve $(\tilde{R}^H V)\beta = -\tilde{R}^H Z$ for $\beta$
     4.12  $P \leftarrow R + (P - \zeta V)\beta$
5    return $(X)$
```

Algorithm 2.2: MEMORY SAVING VERSION(A, M, B, ϵ)

```
1    initial guess $X \in \mathbb{C}^{N\times L}$
2    $R = B - AX$
3    $P = R$
4    choose $\tilde{R} \in \mathbb{C}^{N\times L}$
while  $\max_i(|r^{(i)}|/|b^{(i)}|) \le \epsilon$
do
     4.1   QR decomposition $P = Q\gamma,\ P \leftarrow Q$
     4.2   $U = MP$
     4.3   $V = AU$
     4.4   solve $(\tilde{R}^H V)\alpha = \tilde{R}^H R$ for $\alpha$
     4.5   $R \leftarrow R - V\alpha$
     4.6   $X \leftarrow X + U\alpha$
     4.7   $S = MR$
     4.8   $Z = AS$
     4.9   $\zeta = \mathrm{Tr}(Z_k^H R_k)/\mathrm{Tr}(Z_k^H Z_k)$
     4.10  $X \leftarrow X + \zeta S$
     4.11  $R \leftarrow R - \zeta Z$
     4.12  solve $(\tilde{R}^H V)\beta = -\tilde{R}^H Z$ for $\beta$
     4.13  $P \leftarrow R + (P - \zeta V)\beta$
5    return $(X)$
```

clover term), the matrix A is decomposed as the following 2×2 blocked matrix form,

$$A = \begin{pmatrix} A_{EE} & A_{EO} \\ A_{OE} & A_{OO} \end{pmatrix}, \tag{7}$$

where the subscript E and O denote the even and odd domains, respectively. The SAP preconditioner M_{SAP} is computed as

$$\begin{aligned} M_{SAP} &= K \sum_{j=0}^{N_{SAP}} (1 - AK)^j\,, \\ K &= \begin{pmatrix} B_{EE} & 0 \\ -B_{OO}A_{OE}B_{EE} & B_{OO} \end{pmatrix}, \end{aligned} \tag{8}$$

where B_{EE} (B_{OO}) is an approximation for A_{EE}^{-1} (A_{OO}^{-1}) obtained by SSOR method [11]

$$B_{EE} = (1 - \omega U_{EE})^{-1}\Big[\sum_{j=0}^{N_{SSOR}} (1 - A_{EE}^{SSOR})^j\Big](1 - \omega L_{EE})^{-1}\,, \tag{9}$$

with

$$\begin{aligned} A_{EE}^{SSOR} = \frac{1}{\omega}\Big[&(1 - \omega L_{EE})^{-1} + (1 - \omega U_{EE})^{-1} \\ &+ (\omega - 2)(1 - \omega L_{EE})^{-1}(1 - \omega U_{EE})^{-1}\Big]. \end{aligned} \tag{10}$$

L_{EE} is the forward hopping term and U_{EE} is the backward one. We perform SAP preconditioning in the single precision for effective use of memory bandwidth and network bandwidth between nodes.

It is known that "sloppy" precision can be used in right preconditioning, but not in left one. Suppose calculation of $S = MT$ at line 4.6 in Algorithm 2.1 is performed with "sloppy" precision in k-th iteration. Numerical errors for S_k, Z_k, ζ_k and X_{k+1} may be expressed as

$$S_k \;\to\; S'_k = S_k + \delta S_k\,, \tag{11}$$

$$Z_k \;\to\; Z'_k = AS'_k\,, \tag{12}$$

$$\zeta_k \;\to\; \zeta'_k = \zeta_k + \delta\zeta_k\,, \tag{13}$$

$$X_{k+1} \;\to\; X'_{k+1} = X_k + U_k\alpha_k + \zeta'_k S'_k\,. \tag{14}$$

These yield

$$\begin{aligned} R'_{k+1} &= R_k - V_k\alpha_k - \zeta'_k Z'_k \\ &= R_k - AU_k\alpha_k - \zeta'_k AS'_k \\ &= B - AX_k - A(U_k\alpha_k + \zeta'_k S'_k) \\ &= B - AX'_{k+1}\,, \end{aligned} \tag{15}$$

which satisfies the exact relation between approximate solutions and residuals in $(k+1)$-th iteration. For the case that both $U = MP$ at line 4.2 and $S = MT$ at line 4.6 are computed with "sloppy" precision one can also reproduce the above relation with the following formulae:

$$U_k \rightarrow U'_k = U_k + \delta U_k \,, \tag{16}$$
$$V_k \rightarrow V'_k = AU'_k \,, \tag{17}$$
$$\alpha_k \rightarrow \alpha'_k = \alpha_k + \delta\alpha \,, \tag{18}$$
$$T_k \rightarrow T'_k = R_k - V'_k\alpha'_k \,, \tag{19}$$
$$S_k \rightarrow S''_k = S_k + \delta S \,, \tag{20}$$
$$Z_k \rightarrow Z''_k = AS''_k \,, \tag{21}$$
$$\zeta_k \rightarrow \zeta''_k = \zeta_k + \delta\zeta \,, \tag{22}$$
$$X_{k+1} \rightarrow X''_{k+1} = X_k + U'_k\alpha'_k + \zeta''_k S''_k \,. \tag{23}$$

3. Numerical test

3.1. Choice of parameters

We test modified block BiCGSTAB employing a so-called "local source", $B = [e_1, ..., e_L]$, with $L = 12$ for color-spin components. We use statistically independent 10 configurations generated at almost the physical point, $\kappa_{ud} = 0.137785$ and $\kappa_s = 0.136600$, in 2+1 flavor lattice QCD with the nonperturbatively $O(a)$-improved Wilson quark action and the Iwasaki gauge action [12] at $\beta = 1.9$ on a $32^3 \times 64$ lattice [1]. We choose the hopping parameter $\kappa = 0.137785$ for the Wilson-Dirac equation and $N_{SAP} = 5$ with $8 \times 8 \times 8 \times 8$ domain size for the SAP preconditioning following Ref. [1]. Parameters for SSOR method are also fixed with $N_{SSOR} = 1$ and $\omega = 1.26$. The stopping criterion is set to be $\max_i(|\boldsymbol{r}^{(i)}|/|\boldsymbol{b}^{(i)}|) \leq \epsilon$ with $\epsilon = 10^{-14}$.

3.2. Test environment

Numerical test is performed on 16 nodes of a large-scale cluster system called T2K-Tsukuba. The machine consists of 648 compute nodes providing 95.4Tflops of computing capability. Each node consists of quad-socket, 2.3GHz Quad-Core AMD Opteron Model 8356 processors whose on-chip cache sizes are 64KBytes/core, 512KBytes/core, 2MB/chip for L1, L2, L3, respectively. Each processor has a direct connect memory interface to an 8GBytes DDR2-667 memory and three hypertransport links to connect other processors. All the nodes in the system are connected through a full-bisectional fat-tree network consisting of four interconnection links of 8GBytes/sec aggregate bandwidth with Infiniband. For numerical test we modify the lattice QCD simulation program LDDHMC/ver1.04b12.31 developed by PACS-CS Collaboration [13].

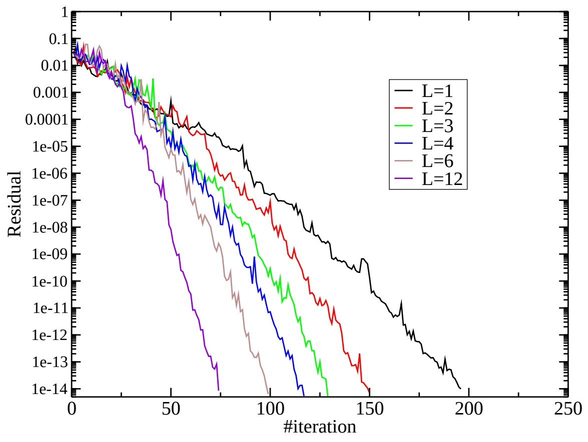

Figure 2: Representative case for residual norm as a function of number of iteration with $L = 1, 2, 3, 4, 6, 12$.

3.3. Results

Figure 2 shows a representative case for residual norm as a function of number of iterations for modified block BiCGSTAB. We observe one of important features of block Krylov subspace methods that the number of iterations required for convergence decreases as the block size L is increased.

$L \times 12/L$	time[s]	T(gain)	NMVM	NM(gain)
1×12	3827(755)	1	17146(3326)	1
2×6	2066(224)	1.9	12942(1379)	1.3
3×4	1619(129)	2.4	10652(832)	1.6
4×3	1145(99)	3.3	9343(835)	1.8
6×2	1040(87)	3.7	7888(663)	2.2
12×1	705(70)	5.4	6106(633)	2.8

Table 1: L dependence for time, gain factor of time, number of matrix-vector multiplication and its gain factor. Central values are given for gain factors.

The results are summarized in Table 1. Second column is total time to solve the Wilson-Dirac equation for all 12 colour-spin components at one local source. In case of $L = 6$, for example, 12 right-hand side vectors are divided into two blocks: $B_1 = [e_1, ..., e_6]$ and $B_2 = [e_7, ..., e_{12}]$. Third column is gain factor of time compared with $L = 1$ case. Fourth and fifth columns are number of matrix-vector multiplication (NMVM) and its gain factor, respectively. Modified block BiCGSTAB with $L = 12$ is about 5 times faster than $L = 1$ case. The iteration number is reduced by a factor of three. Additional speed-up by a factor of two is obtained by cache tuning. This is roughly consistent with the enhancement of Flops/Byte from 1.05 at $L = 1$ to 1.95 at $L = 12$ plotted in Fig. 1.

4. Conclusions

In this paper, we have carried out numerical test for block BiCGSTAB with two modifications of the QR decomposition and the SAP preconditioner in lattice QCD at almost physical quark masses. The QR decomposition successfully removes the problem in block BiCGSTAB that is the deviation between the true and the recursive residuals. We find remarkable cost reduction at large L due to smaller number of iterations and efficient cache usage. Further gain could be expected in calculations of disconnected diagram and reweighting factor, where larger value of L is required. One concern is that numerical cost for modified Gram-Schmidt method increases as $O(L^2)$.

Acknowledgments

Numerical calculations for the present work have been carried out on the T2K-Tsukuba computer at the University of Tsukuba. This work is supported in part by Grants-in-Aid for Scientific Research from the Ministry of Education, Culture, Sports, Science and Technology (Nos. 20800009, 20105002, 22244018).

References

[1] PACS-CS Collaboration, S. Aoki et al., Physical Point Simulation in 2+1 Flavor Lattice QCD, Phys. Rev. D 81 (2010) 074503.

[2] D. P. O'Leary, The block conjugate gradient algorithm and related methods, Linear Algebra Appl., 29 (1980), pp. 293-322; A. A. Nikishin and A. Y. Yeremin, Variable block CG algorithms for solving large sparse symmetric positive-definite linear systems on parallel computers, SIAM J. Matrix Anal. Appl., 16 (1995), No. 4, pp. 1135-1153

[3] A. El Guennouni, K. Jbilou, H. Sadok, A Block Version of BiCGSTAB for Linear Systems with Multiple Right-Hand Sides, Elec. Trans. Numer. Anal. 16 (2003) 129.

[4] H. Tadano, T. Sakurai, Y. Kuramashi, Block BiCGGR: A New Block Krylov Subspace Method for Computing High Accuracy Solutions JSIAM Lett. 1 (2009) 44.

[5] T. Sakurai, H. Tadano, Y. Kuramashi, Application of block Krylov subspace algorithms to the Wilson-Dirac equation with multiple right-hand sides in lattice QCD Comput. Phys. Commun. 181, (2010) 113.

[6] H. Tadano, Y. Kuramashi, T. Sakurai, Application of preconditioned block BiCGGR to the Wilson-Dirac equation with multiple right-hand sides in lattice QCD, Comput. Phys. Commun. 181, (2010) 883.

[7] A. A. Dubrulle, Retooling the method of block conjugate gradients, Elec. Trans. Numer. Anal. 12 (2001) 216.

[8] H. Tadano et al., in preparation.

[9] M. Lüscher, Lattice QCD and the Schwarz alternating procedure, JHEP 0305, 052 (2003); Comput. Phys. Commun. 165, (2005) 119.

[10] S. Dürr et al., Scaling study of dynamical smeared-link clover fermions, Phys. Rev. D 79 (2009) 014501.

[11] S. Fischer et al., A Parallel SSOR Preconditioner for Lattice QCD, Comput. Phys. Commun. 98 (1996) 20, hep-lat/9602019.

[12] Y. Iwasaki, Renormalization group analysis of lattice theories and improved lattice action; 2, four-dimensional non-Abelian SU(N) gauge model, preprint, UTHEP-118 (Dec. 1983), unpublished.

[13] PACS-CS Collaboration, S. Aoki et al., 2+1 Flavor Lattice QCD toward the Physical Point, Phys. Rev. D 79 (2009) 034503.